The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

グラフ上の問題に対する難しいインスタンスの自動生成

Learning to Find Hard Instances of Graph Problems

佐藤 竜馬 *1　山田 誠 *1*2*3　鹿島 久嗣 *1*2
Ryoma Sato　Makoto Yamada　Hisashi Kashima

*1京都大学 Kyoto University　*2理化学研究所 革新知能統合研究センター RIKEN Center for AIP　*3JST さきがけ JST PRESTO

Finding *hard instances*, which needs a long time to solve, of graph problems is important for building a good benchmark for evaluating the performance of algorithms and analyzing algorithms to accelerate algorithms. In this paper, We aim at automatically generating hard instances of graph problems. We formulate finding hard instances of graph problems as an optimization problem and propose a method to automatically find hard instances by solving the optimization problem. The advantage of the proposed algorithm is that it does not require any task-specific knowledge. To the best of our knowledge, it is the first non-trivial method in the literature to automatically find hard instances by using optimization. Through experiments on various problems, we show that our proposed method can generate a few to several orders of magnitude harder instances than the random based approach in many settings, and especially our method outperforms rule-based algorithms in the 3-coloring problem.

1. はじめに

組合せ問題のアルゴリズムを与えられたとき，どのようにしてそのアルゴリズムが解くのに時間がかかるインスタンスを自動で見つけることができるだろうか？ある与えられたアルゴリズムにとって解決に長い時間がかかるインスタンスのことを**難しいインスタンス**と呼ぶ。難しいインスタンスを見つけることは，次の点で重要である。

理由 1 アルゴリズムの分析と高速化に役立つ。

理由 2 ベンチマークとして使うことができる。難しいインスタンス を含むベンチマークを生成することは，アルゴリズムを評価することにとって重要である。

難しいインスタンスを見つけるための最も単純な方法は，ランダムに多数のインスタンスを生成し，それらを評価し，それらのうち最も難しいインスタンスを出力することである。この手法は多くの問題について適用でき，また単純であるが，多くのアルゴリズム（*e.g.,* クイックソート）は最悪計算量を実現するインスタンスに比べて，ランダムなインスタンスをはるかに速く処理できるので，この方法によって難しいインスタンスを発見するのは効率的ではない。難しいインスタンスを効率的に見つけるためには，問題の構造をとらえることができる方法を開発する必要がある。本研究ではグラフ問題に焦点をあて，グラフ問題の難しいインスタンスを自動でかつ効率よく生成するアルゴリズムの開発を目指す。グラフ問題は，彩色問題，最小頂点被覆問題，最大クリーク問題など，理論上，応用上共に重要な問題を数多く含むため，自動的にグラフ問題の難しいインスタンスを生成できるアルゴリズムの提案は大きな貢献となる。本研究では，グラフ問題の難しいインスタンスを見つける問題を最適化問題とし，それを解決するために山登り法，疑似焼きなまし法，そしてニューラルネットワークと強化学習を使用することを提案する。本研究における提案手法は広範囲の問題に適用可能であり，一様ランダムグラフモデルに比べてはるかに難しいインスタンスを生成することができる。

本研究の主な貢献は以下のようにまとめられる。

連絡先: 佐藤竜馬，京都大学，r.sato@ml.ist.i.kyoto-u.ac.jp

1. **定式化** グラフ問題の難しいインスタンスを見つける問題を最適化問題として定式化する。
2. **新手法の提案** 最適化問題を解くことによって困難なインスタンスを見つけるための一般的で効果的な方法を提案する。
3. **有効性** 4 つの問題と 6 つのアルゴリズムを用いた実験を通して，本手法の有効性を示す。

2. 提案手法

本章では，まず問題設定を定義する。次に，難しいインスタンスを生成する問題を最適化問題として定式化することを提案し，ニューラルネットワークと強化学習に基づく難しいインスタンス生成アルゴリズムを提案する。

2.1 問題設定

本節では，本研究で取り組むグラフ問題の難しいインスタンスを見つける問題を定義する。本研究では，グラフ彩色問題，最小頂点被覆問題，最大クリーク問題などの多くの重要な問題を含む，無向・重みなし・単純グラフを取り扱う。グラフ彩色問題，最小頂点被覆問題，最大クリーク問題はスケジューリング問題やコンパイラのレジスタ割り当て問題やコミュニティ検出など，多くの応用上の問題に登場するため，ランダムなインスタンスを含む多くのインスタンスを現実的な時間内に処理することができる様々なアルゴリズムが提案されている。一方で，これらの問題は NP 困難問題であるので，そのような効率的なアルゴリズムも最悪ケースでの計算量は指数的に増加するはずである。そのようなケースを見つけることは，第 1 章で述べたように，アルゴリズムを解析し，高速化することにとって重要である。

本研究の目標は，グラフ問題に対するアルゴリズムが与えられたとき，そのアルゴリズムが解くために多くのステップを必要とするインスタンス（*i.e.,* 難しいインスタンス） を自動で発見することである。特に，問題固有の特性や専門家による洞察を用いず，各インスタンスの難しさの値のみを使用して難しいインスタンスを生成する。インスタンスの難しさの定義は任

意であるが，そのインスタンスに対して実際にアルゴリズムを実行することによって計算できるものとする。例えば，実験では，三彩色問題のインスタンスの難しさの値は，Brélaz のアルゴリズムの再帰呼び出しの数によって定義され，グラフ同型判定問題のインスタンス難しさは， Nauty [McKay 14] が問題を解決するのに必要な時間によって定義される。このように定義することで，インスタンスの難しさを定量的に評価できるようになると共に，最適化問題としての定式化が可能となる。アルゴリズム L を用いたときのインスタンス x の難しさの値を $\mathrm{hardness}(x, L)$ として表記する。

仮定 1 （小さいインスタンス）： 単に頂点の数を増やすことでインスタンスを任意に難しくすることができるが，このようにして生成されたインスタンスは自明であり，このような手法は実用的ではない。 また，小さなインスタンスは可視化することができ，解釈や分析が容易である。 本研究では，出力するインスタンスのサイズを小さな値で固定し，そのサイズの中で難しいインスタンスを生成することを目指す。

仮定 2 （サンプル効率）： 難しさの評価は一般にアルゴリズムをシミュレートする必要があるため時間がかかり，インスタンスが難しい場合は特に時間がかかる。したがって，あまりにも多くのインスタンスを評価することは現実的ではないため，難しいインスタンスをより効率的に見つけることが重要である。本研究では，評価回数の上限を B に設定する。 この制約により，あまりにも多くの目的関数を評価する方法（ *e.g.*, brute-force search，遺伝的アルゴリズム）は使用できない。

2.2 最適化問題としての定式化

本研究では，難しいインスタンスを見つけるタスクを最適化問題として定式化し，直接最適化することを提案する。難しいインスタンス を見つけるタスクを単純に定式化すると次のように書くことができる。

Problem 1

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & \mathrm{hardness}(x, L) \\ \text{subject to} \quad & x \text{ is an instance of } Q. \end{aligned}$$

ここで Q は問題で， L は与えられたアルゴリズムである。x が Q のインスタンスであるという条件は一般には Q に依存し，書き下すことはできないが，本研究では無向・重みなし・単純グラフ上の問題を考えているため， x が Q のインスタンスであるときかつそのときのみ，隣接行列 $A \in \{0,1\}^{n(n-1)/2}$ として表すことができる。したがって， Problem 1 は以下のようにより具体的に制約を書き直すことができる。

Problem 2

$$\begin{aligned} \underset{A}{\text{maximize}} \quad & \mathrm{hardness}(A, L) \\ \text{subject to} \quad & A \in \{0,1\}^{n(n-1)/2}. \end{aligned}$$

本研究でははじめに，山登り法と疑似焼きなまし法を使用して問題 2 を解くことを考える。山登り法と疑似焼きなまし法はよく知られたメタヒューリスティック探索アルゴリズムである。1 つのグラフ A を状態とし， A から 1 つの辺を削除するか， 1 つの辺を追加することで，近傍解を生成する。近傍の数が多いため，それらすべての難しさを計算することはサンプル効率の観点から効率的ではない。例えば，頂点数が $n = 50$ の場合，グラフの近傍数は $n(n-1)/2 = 1225$ となる。この問題に対処するために，本研究では乱択アルゴリズムを採用する。まず，最初に現在のグラフ A の近傍 N を無作為に選び，その難しさの値を評価する。N の難しさが A よりも高い場合は， N を次の状態として遷移し，そうでなければ現在の状態 A に留まる。

以上の手法により， Problem 2 を解くことができるが， A は離散変数であるため，Problem 2 を最適化するのは困難である。そこで，本研究では最適値を変えずにこの問題を連続空間上の問題に変換することを提案する。

Problem 3

$$\begin{aligned} \underset{P}{\text{maximize}} \quad & \mathbb{E}_{A \sim \mathrm{Bernoulli}(P)}[\mathrm{hardness}(A, L)] \\ \text{subject to} \quad & P \in [0,1]^{n(n-1)/2} \end{aligned}$$

ここで， $A \sim \mathrm{Bernoulli}(P)$ は，各 A_i が $\mathrm{Bernoulli}(P_i)$ （ベルヌーイ分布）から独立してサンプリングされることを表す。この変換によって最適値が変わることはない。このことを次に示す。

Theorem 1. 問題 2 と 問題 3 の最適値は同じである。

Proof. M を問題 2 の最適値とし，$\mathrm{hardness}(A^*, L) = M$ とする。 M' を問題 3 の最適値とする。M は問題 2 の最適値であるので，任意のグラフ A について $\mathrm{hardness}(A, L) \leq M$ が成り立つ。 したがって，任意の P に対して，$\mathbb{E}_{A \sim \mathrm{Bernoulli}(P)}[\mathrm{hardness}(A, L)] \leq M$ となる。よって $M' \leq M$。 一方，$M' \geq \mathbb{E}_{A \sim \mathrm{Bernoulli}(A^*)}[\mathrm{hardness}(A, L)] = \mathrm{hardness}(A^*, L) = M$ となる。したがって，$M = M'$ が成り立つ。 □

Theorem 2. 問題 3 の目的関数

$$f(P) = \mathbb{E}_{A \sim Bernoulli(P)}[hardness(A, L)]$$

は解析的関数である。特に， C^∞ 級関数である。

Proof.

$$\begin{aligned} f(P) &= \mathbb{E}_{A \sim \mathrm{Bernoulli}(P)}[\mathrm{hardness}(A, L)] \\ &= \sum_{A} \prod_{i=1}^{n(n-1)/2} (P_i^{A_i}(1-P_i)^{1-A_i}) \cdot \mathrm{hardness}(A, L) \end{aligned}$$

である。ここで A は $A \in \{0,1\}^{n(n-1)/2}$ をわたる。 $f(P)$ は P の要素についての多項式であるので， $f(P)$ は C^∞ 級関数であり，解析的関数である。 □

Theorem 2 より，問題 3 の目的関数が滑らかであるため，連続空間で効率的に探索ができることが分かる。

2.3 確率的グラフモデル

本研究では，確率グラフ生成モデルと即時強化学習を使用した機械学習的なアプローチを使用して，問題 3 を解くことを提案する。他のモデルの選択肢としては，ノードやエッジを 1 つずつ出力する逐次モデルが考えられるが，特にグラフが大きい場合，そのようなモデルは訓練するのが難しい [Ma 18] ため，この方法は採用しない。即時強化学習の枠組みでは，エージェントの行動は難しいと予想されるインスタンス A に対応し，そのアクションの報酬 r はアルゴリズムを A に対して実

行してたときのコストである（*i.e.*, $r = \text{hardness}(A, L)$）。報酬 r は，報酬の予測精度を向上させるために使用される。環境からの入力は存在しないため，入力としてノイズ z を使用する。エピソードに含まれるアクションは 1 つだけなので，各アクションは独立である。

行動を決定するために，ニューラルネットワークモデルは，各辺 i に対して辺 i が現れる確率 P_i を出力する（$i = 1, 2, \ldots, n(n-1)/2$ ）。そして，確率 P_i に従って各辺を独立にサンプリングし，グラフ A を構築する（*i.e.*, アクション）。次に，報酬を得るために $\text{hardness}(A, L)$ を評価する。最後に，REINFORCE アルゴリズム [Williams 92] を使ってニューラルネットワークモデル w_i の重みを更新する：

$$w_i = w_i + \alpha r \frac{\partial}{\partial w_i} \sum_{i=1}^{n(n-1)/2} (\log P_i^{A_i} + \log(1 - P_i)^{(1-A_i)})$$

ここで α は学習率である。 P が観測されている場合， A_i と A_j $(i \neq j)$ は独立しているが，条件が無い場合は P は独立していないため A_i と A_j $(i \neq j)$ は独立ではなく，提案手法はエッジ間の非線形な関係もモデル化できる。

3. 拡張

本研究での提案手法では，難しさの値の選択は任意である。したがって，提案手法は，解決するのに長い時間を必要とするインスタンスだけでなく，他の意味で難しいインスタンスも見つけることができる。本章では応用上重要な二つの拡張について述べる。

3.1 近似度の推定

L を近似アルゴリズム，A を問題のインスタンス，$L(A)$ を L が A に対して出力する値，$\text{OPT}(A)$ を A の最適値とする。$\text{OPT}(A)$ は厳密アルゴリズムに A を入力することで計算できる。L の近似度は，最小化問題においては

$$r(L) = \max_{A \text{ is an instance}} \frac{L(A)}{\text{OPT}(A)}$$

および最大化問題においては

$$r(L) = \max_{A \text{ is an instance}} \frac{\text{OPT}(A)}{L(A)}$$

で定義される。 近似度の推定は，近似アルゴリズムの性能を調べるために重要である。 しかし， $\frac{L(A)}{\text{OPT}(A)}$ および $\frac{\text{OPT}(A)}{L(A)}$ を最大化するインスタンスを発見することは一般に難しい。そこで，インスタンス A の難しさの値として $\frac{L(A)}{\text{OPT}(A)}$ および $\frac{\text{OPT}(A)}{L(A)}$ を使用することで提案手法を用いて最大値に近い値をもつインスタンスを発見することができる。

3.2 列挙アルゴリズムの計算量

列挙アルゴリズムはある特性を満たすすべての要素を出力するアルゴリズムである。列挙アルゴリズムの計算効率を評価するときは，全ての要素の列挙にかかった合計時間だけではなく，アルゴリズムが各要素を出力する最大遅延時間 (maximum delay) やならし計算時間 (amortized time) がしばしば使用される。 合計時間の代わりに最大遅延時間やならし計算時間をインスタンスの難しさの値として使用することで本研究における提案方法は最大遅延時間やならし計算時間の意味で難しいインスタンスを生成することもできる。

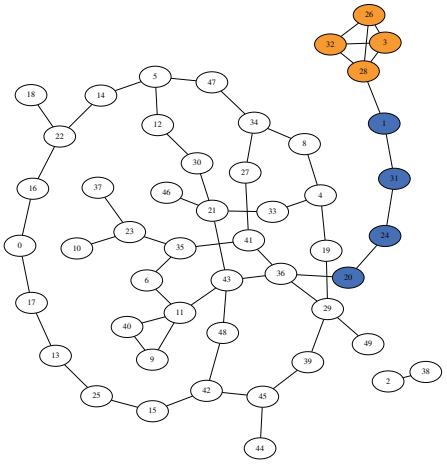

図 1: ニューラルネットワークモデルが生成した三彩色問題のインスタンスの例。 バックトラッキング探索アルゴリズムは，このインスタンスを処理するために十億回以上の再帰呼び出しを必要とする。 ノード 3，26，28，および 32 で構成される部分グラフ（図中右上端，オレンジ色）は 4 クリークを形成するため，このインスタンスは三彩色可能ではない。この 4 クリークがパス（図中青色）で大きな連結成分に接続していることが，このインスタンスを難しくしている原因である。

4. 実験

ニューラルネットワークモデルとしては 4 層の多層パーセプトロンを用いる。Adam [Kingma 14] を用いて学習を行い，学習率を 0.001，β_1 を 0.9，β_2 を 0.999 に設定する。自明なベースラインとして一様ランダムグラフモデル（Erdős-Rényi モデル）による生成と評価を繰り返す手法を用い，強力なベースラインとして三彩色問題およびグラフ同型問題でいくつかのルールベースアルゴリズムを用いる。これらのベースラインでは，アルゴリズムを使用して B（*i.e.*, 評価数の上限）個のグラフが生成され，その中で最も難しいインスタンスが出力される。$B = 100000$ に設定し，いずれの手法も 3 日以上経過すると強制的に停止させる。問題に応じて，出力グラフのノード数は基本的に $n = 50$ ノードに固定するが， $n = 50$ ノードを用いたときに多くの手法が非常に難しいインスタンスを生成して比較が困難になる場合に $n = 32$ ノードに固定する。対象となる問題とアルゴリズム以下のものを用いる。

三彩色問題 (3-coloring)：アルゴリズムとしては，Brélaz のヒューリスティックアルゴリズム [Brélaz 79] に基づくバックトラッキング探索 (Brélaz) を使用し，難しさの値は再帰回数とする。頂点数は $n = 50$ とする。この問題では，ベースラインとしてルールベースのアルゴリズムも使用する。

最小頂点被覆問題 (Vertex Cover)：アルゴリズムとしては，極大マッチングによる上限を用いて，制約の数をより早く減らすことができる頂点を探索する分枝限定法 (B&B) を使用し，難しさの値は再帰回数とする。頂点数は $n = 50$ とする。

最大クリーク問題 (Clique)： アルゴリズムとしては，Bron-Kerbosch のアルゴリズム (BK)，ピボットを用いない Bron-Kerbosch のアルゴリズム (BKNP), Fast Max-Cliquer [Pattabiraman 13] (FMC) を使用し，難しさの値はそれぞれ再帰回数，再帰回数， CPU 時間（10^{-6} 秒）とする。頂点数は $n = 32$ とする。

表 1: 実験結果：各値は， 5 回の試行を行い，それぞれの試行で生成した最も難しいインスタンスの難しさを平均したものである。

問題	3-coloring	Vertex Cover	Clique			Isomorphism
アルゴリズム	Brélaz	B&B	BK	BKNP	FMC	Nauty
ニューラルネットワーク	**1151770980.6**	12771.8	74499.6	**4706426.4**	**5882966.0**	**9400.0**
山登り法	212276998.8	**46659.8**	**160091.8**	2615672.0	195117.4	120.6
焼きなまし法	602.2	8003.2	12024.0	2518357.4	147054.4	68.4
Erdős-Rényi $p = 0.1$	937.4	761.0	86.8	105.8	27966.0	122.2
Erdős-Rényi $p = 0.5$	2.0	635.6	562.4	1503.6	36657.6	32.6
Erdős-Rényi $p = 0.9$	2.0	489.8	9365.4	327251.4	107445.0	118.0
[Cheeseman 91]	1567.6	N/A	N/A	N/A	N/A	N/A
[Hogg 94]	31708.2	N/A	N/A	N/A	N/A	N/A
[Vlasie 95]	85353.6	N/A	N/A	N/A	N/A	N/A
[Mizuno 08]	219342.2	N/A	N/A	N/A	N/A	N/A
$R(B(G_n, \sigma))$ [Neuen 17]	N/A	N/A	N/A	N/A	N/A	2700.0
$R^*(B^*(G_n, \sigma))$ [Neuen 17]	N/A	N/A	N/A	N/A	N/A	2182.0

グラフ同型問題 (Isomorphism)： アルゴリズムとしては，Nauty [McKay 14] を使用し，難しさの値は CPU 時間（10^{-7} 秒）とする。頂点数は $n = 50$ とする。この問題では，ベースラインとしてルールベースのアルゴリズムも使用する。

実験結果を表 1 に示す。どのアルゴリズムに対しても，ニューラルネットワークモデルと山登り法が一貫して Erdős-Rényi モデルよりも難しいインスタンスを生成できている。特に，三彩色問題においては，ルールベースの難しいインスタンス生成アルゴリズムに比べてもはるかに難しいインスタンスを生成できている。ニューラルネットワークモデルが生成した難しいインスタンスの例を 1 に示す。

5. 結論

本論文では，グラフ問題の難しいインスタンスを最適化問題として定式化した（問題 2，3）。 そして，これらの問題を解決するために，山登り法，疑似焼きなまし法，ニューラルネットワークと強化学習という三つの方法を提案した。 実験では，さまざまな問題とアルゴリズムを用い，ニューラルネットワークと強化学習による手法が多くの設定において Erdős-Rényi モデルより一桁から数桁難しいインスタンスを見つけることができることを示した。 特に三彩色問題では，提案手法はルールベースのアルゴリズムよりも難しいインスタンスを発見することができた（表 1）。

謝辞

本研究は JSPS 科研費 15H01704 および JST PRESTO JPMJPR165A の助成を受けた。また，提案手法の拡張についての実りある議論を交わして頂いた京都大学の小林靖明助教と NII の Alessio Conte 特任研究員に深く感謝の意を表する。

参考文献

[Brélaz 79] Brélaz, D.: New Methods to Color Vertices of a Graph, *Commun. ACM*, Vol. 22, No. 4, pp. 251–256 (1979)

[Cheeseman 91] Cheeseman, P. C., Kanefsky, B., and Taylor, W. M.: Where the Really Hard Problems Are, in *IJCAI*, pp. 331–340 (1991)

[Hogg 94] Hogg, T. and Williams, C. P.: The Hardest Constraint Problems: A Double Phase Transition, *Artif. Intell.*, Vol. 69, No. 1-2, pp. 359–377 (1994)

[Kingma 14] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *CoRR*, Vol. abs/1412.6980, (2014)

[Ma 18] Ma, T., Chen, J., and Xiao, C.: Constrained Generation of Semantically Valid Graphs via Regularizing Variational Autoencoders, in *NeurIPS*, pp. 7113–7124 (2018)

[McKay 14] McKay, B. D. and Piperno, A.: Practical graph isomorphism, II, *Journal of Symbolic Computation*, Vol. 60, No. 0, pp. 94 – 112 (2014)

[Mizuno 08] Mizuno, K. and Nishihara, S.: Constructive generation of very hard 3-colorability instances, *Discrete Applied Mathematics*, Vol. 156, No. 2, pp. 218–229 (2008)

[Neuen 17] Neuen, D. and Schweitzer, P.: Benchmark Graphs for Practical Graph Isomorphism, in *ESA*, pp. 60:1–60:14 (2017)

[Pattabiraman 13] Pattabiraman, B., Patwary, M. M. A., Gebremedhin, A. H., Liao, W., and Choudhary, A. N.: Fast Algorithms for the Maximum Clique Problem on Massive Sparse Graphs, in *Algorithms and Models for the Web Graph, WAW*, pp. 156–169 (2013)

[Vlasie 95] Vlasie, R. D.: Systematic generation of very hard cases for graph 3-colorability, in *ICTAI*, pp. 114–119 (1995)

[Williams 92] Williams, R. J.: Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning, *Mach. Learn.*, Vol. 8, No. 3-4, pp. 229–256 (1992)